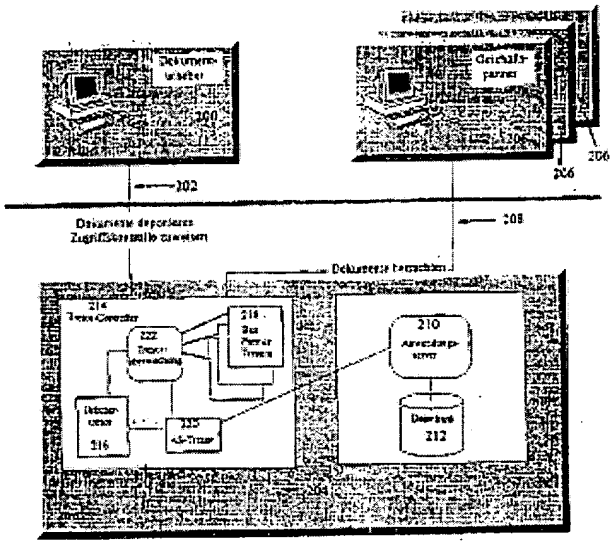
System for electronic data archiving

Patent number: DE19960978
Publication date: 2000-08-03
Inventor: MIRLAS LEV (CA); TCHAO SUNG WEI (CA); BACHA HAMID (US); CARROLL ROBERT BRUCE (US)
Applicant: IBM (US)
Classification:
- international: G06F17/30; H04L9/32; G06F12/14
- european:
Application number: DE19991060978 19991217
Priority number(s): CA19982256936 19981223

Also published as:
CA2256936 (A1)

Abstract of **DE19960978**

System for controlling electronic data access where a data archive is controlled by a third party and a first party deposits data files in an archive that can then be searched by an authorized second party.- DETAILED DESCRIPTION - Secure system comprises: a first agent program that allows a computer to deposit a data file in a data archive, a second agent program that allows a user computer access rights to the electronic data file in the archive. An authorization list for the data file is produced and managed by the provision (first) agent program. An access right list can be accessed using the second (user) agent program. Means are provided to allow the provider computer to send access rights to the user computer or to alter its access rights.Means are also provided to allow the provider computer to check the access rights of the user computer before it has free access to the data file using the second agent program



(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 199 60 978 A 1**

(51) Int. Cl.[7]:
G 06 F 17/30
H 04 L 9/32
G 06 F 12/14

(21) Aktenzeichen: 199 60 978.0
(22) Anmeldetag: 17. 12. 1999
(43) Offenlegungstag: 3. 8. 2000

DE 199 60 978 A 1

(30) Unionspriorität:
2256936 23. 12. 1998 CA

(71) Anmelder:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Duscher, R., Dipl.-Phys. Dr.rer.nat., Pat.-Ass., 71034 Böblingen

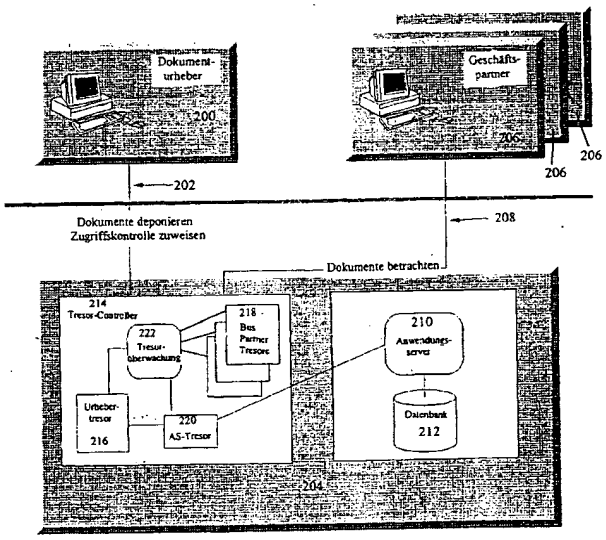
(72) Erfinder:
Bacha, Hamid, Great Falls, Va., US; Carroll, Robert Bruce, Mount Kisco, N.Y., US; Mirlas, Lev, Thornhill, Ontario, CA; Tchao, Sung Wei, Toronto, Ontario, CA

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) System für ein elektronisches Datenarchiv mit Erzwingung einer Zugriffskontrolle beim Suchen und Abrufen von Daten

(57) Wenn ein elektronisches Dokument zur Revision durch andere Stellen verfügbar gemacht wird, ist es oft vorteilhaft, das Dokument in einem von einer dritten Partei verwalteten Archiv oder einer von einer dritten Partei verwalteten Datenbank zu speichern. Es wird ein System zur Verfügung gestellt, in dem die Stellen, die vom Urheber eines Dokuments die Berechtigung zum Zugriff auf dieses Dokument im Datenarchiv erhalten haben, sicher nach dem im Archiv einer dritten Partei aufbewahrten Dokument suchen können, ohne dass sie darauf vertrauen müssen, dass der Verwalter des Archiv sichere Information über ihre Zugriffsrechte liefert. Der Dokumenturheber, der Archivverwalter und alle Stellen, die Zugriffsrechte auf Daten im Archiv besitzen, haben Tresorumgebungen, die sichere Erweiterungen ihrer betreffenden Arbeitsbereiche sind. Der Tresor des Dokumenturhebers verwaltet für jedes im Archiv deponierte Dokument eine Zugriffskontroll-Liste (ACL). Der Tresor jeder Stelle, die Zugriffsrechte auf Dokumente im Archiv besitzt, verwaltet eine Fähigkeitsliste der Zugriffsrechte der betreffenden Stelle auf alle im Archiv gespeicherten Dokumente. Die Stelle selber bewahrt auf ihrem eigenen Arbeitsplatz einen Beweis der neuesten Version ihrer Fähigkeitsliste auf. Wenn die ACL für ein Dokument im Tresor des Dokumenturhebers aktualisiert wird, stellt der Tresor des Urhebers fest, welche Stellen von der Änderung betroffen sind, und überträgt die Änderungen an die Tresore der betroffenen ...



DE 199 60 978 A 1

Beschreibung

Gegenstand der Erfindung

Die vorliegende Erfindung betrifft das Gebiet der elektronischen Datenspeicherung und liefert speziell ein sicheres Datenarchiv- und -austauschsystem, das von einer dritten Partei, die die Funktion eines Verwalters ausübt, verwaltet wird, und in dem eine Zugriffskontrolle beim Suchen und Abrufen von Daten erzwungen wird.

Hintergrund der Erfindung

Neuere parallele Fortschritte in der Netzwerkkommunikation und der PKI-Technologie (public key infrastructure – Infrastruktur öffentlicher Schlüssel) haben bewirkt, dass Unternehmen und Institutionen beginnen, elektronische Dokumentation zur Aufzeichnung und für Transaktionen jeglicher Art einzusetzen. Mit Verbesserungen bei der Integrität und Sicherheit der Übertragung kann zuversichtlich davon ausgegangen werden, dass Dokumente, die elektronisch über das Internet und andere offene Netzwerke gesendet werden, intakt und unverfälscht ankommen. Datenbankverwaltungssysteme, die mit modernen Computerspeichern mit einer Kapazität von mehreren Gigabyte gekoppelt sind, haben es Unternehmen und Institutionen ermöglicht, auf die Aufbewahrung von Dokumenten in Papierform zu verzichten, deren Masse Immobilienkosten verursacht.

Typischerweise müssen Daten, die von einer Stelle stammen, aus verschiedenen Gründen an eine andere übertragen werden, z. B. zur Aufbewahrung, zur Prüfung usw. Die Datenelemente könnten in Form unstrukturierter Dokumentdateien oder strukturierter Datensätze vorliegen wie z. B. Konto- und andere Finanzinformationen. Im Beispiel unstrukturierter Daten kann es notwendig sein, ein Dokument zum Zweck der Prüfung vom Ursprungssystem an andere Computer im gleichen System oder an Computer auf anderen Systemen zu schicken. Dies könnte gleichermaßen in einer Geschäftssituation (z. B. einem Vorschlag für ein Joint Venture oder einer komplexen Angebotsausschreibung) wie auch in einer Institution (z. B. wenn eine Dissertation von akademischen Beratern überprüft wird, bevor sie einer Prüfungskommission vorgelegt wird) vorkommen. Das Dokument ist elektronisch erstellt worden, da auf diese Weise Überarbeitungen und Einfügungen (speziell wenn sie umfangreich sind) leicht eingearbeitet werden können, ohne dass jedesmal das gesamte Dokument neu getippt werden muß.

Wenn das Dokument in elektronischer Form vorliegt, kann es auch leichter überprüft werden, weil es in dieser Form leichter zu übertragen ist. Vorgesehene Betrachter können feststellen, dass ein Dokument verfügbar ist, indem sie das System durchsuchen, nachdem ihnen der Zugriff auf den Speicherort des Dokuments gewährt worden ist.

Es gibt mehrere Gründe, z. B. Sicherheit, Datenintegrität und System- oder Netzwerkverfügbarkeit, weshalb der Dokumenturheber ein Dokument nicht lokal speichern will, wenn dies bedeutet, dass hinter der Firewall Dritten Zugriff gewährt wird. Diese Gründe werden in unserer gleichzeitig eingereichten Patentanmeldung mit dem Titel "System for Electronic Repository of Data Enforcing Access Control on Data Retrieval" (IBM Docket No. CA998-030), das gemeinsam übertragen wurde und hiermit durch Bezugnahme des vorliegenden Dokuments ist, ausführlicher beschrieben.

Unsere gleichzeitig eingereichte Anmeldung betrifft ein System, in dem die Integrität der und der Zugriff auf die in einem Archiv gespeicherten Daten unabhängig von Aktionen der als Verwalter des Archivs agierenden dritten Partei verwaltet wird.

Die in der genannten Anmeldung beschriebene Erfindung ist bei Systemen mit einer großen Anzahl von Dokumenten, die für eine große Anzahl von Benutzern zugänglich sind, sehr effizient, da die Information über den autorisierten Zugriff auf die Dokumente an einer einzigen, zentralen Stelle gespeichert werden, und zwar im Archiv selber. Benutzer erhalten durch systemexterne Mittel sichere Kenntnis ihres Zugriffs auf Dokumente.

Die vorliegende Erfindung ist eine Abwandlung, in der das System selber die Information über den autorisierten Zugriff enthält, die auch sicher vor Aktionen der als Verwalter des Archivs agierenden dritten Partei ist.

Kurzbeschreibung der Erfindung

Es ist deshalb eine Aufgabe der vorliegenden Erfindung, ein System zur elektronischen Speicherung und zum elektronischen Austausch von Dokumenten zur Verfügung zu stellen, in dem die Dokumente physisch in einem von einer dritten Partei verwalteten Archiv gespeichert werden, in dem die Benutzer aber suchen können, um festzustellen, auf welche Dokumente im Archiv sie zugreifen können.

Eine weitere Aufgabe der Erfindung besteht darin, ein System zur Verfügung zu stellen, in dem die Integrität der im Archiv gespeicherten Informationen über autorisierte Zugriffe im System verfügbar, aber nicht von Aktionen der dritten Partei, die das Archiv verwaltet, abhängig ist.

In einem Aspekt hat die vorliegende Erfindung also ein sicheres System zum Suchen elektronischer Datendateien in einem Datenarchiv zum Ziel. Das System besteht aus einer Kommunikationsumgebung, in der ein erstes Agentenprogramm für einen Computer, der eine elektronische Datendatei im Datenarchivsystem deponiert, und ein zweites Agentenprogramm für einen ersten Benutzercomputer mit Zugriffsrecht auf die elektronische Datendatei vorhanden ist. In einer Nachweisliste für die elektronische Datendatei sind Zugriffskontrollen für die elektronische Datendatei aufgeführt. Die Nachweisliste ist für ein erstes Agentenprogramm zugänglich und wird von diesem verwaltet. Der erste Benutzercomputer besitzt eine Aufzeichnung seiner Zugriffsrechte auf die elektronische Datendatei, die für das zweite Agentenprogramm zugänglich ist und von diesem verwaltet wird. Wenn an der Nachweisliste Änderungen vorgenommen werden, die die Zugriffsrechte des ersten Computers auf die elektronische Datendatei betreffen, werden diese Änderungen vom ersten Agentenprogramm an das zweite Agentenprogramm übertragen, so dass die Aufzeichnung des ersten Benutzercomputers über seine Zugriffsrechte aktualisiert werden kann. Das erste Agentenprogramm ist auch in der Lage, die Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei zu prüfen, bevor die elektronische Datendatei für das zweite Agentenprogramm freigegeben wird.

In einem weiteren Aspekt bietet die Erfindung ein Verfahren für eine sichere elektronische Datensuche in einem elektronischen Datenarchiv in einem System mit einer Nachweisliste, in der Zugriffsrechte auf die elektronische Datendatei im Datenarchiv aufgeführt sind, und einer Aufzeichnung, in der Dokumentzugriffsrechte für jeden Computer mit Zugriff auf die im Archiv gespeicherten elektronischen Daten aufgeführt sind. Das Verfahren besteht aus der Aktualisierung einer Nachweisliste für eine im Archiv gespeicherte elektronische Datendatei, der Identifikation aller von der Aktualisierung betroffenen Computer mit geändertem Zugriffsrecht auf die elektronische Datendatei, die Übertragung der Änderung des Zugriffsrechts an alle betroffenen Computer, die Aktualisierung der Zugriffsrecht-Aufzeich-

nungen aller betroffenen Computer und die Übertragung der aktualisierten Zugriffsrecht-Aufzeichnungen an die betroffenen Computer.

In einem weiteren Aspekt bietet die Erfindung ein sicheres System zum Suchen in einem Datenarchivsystem gespeicherter elektronischer Daten, das Mittel besitzt, um einen Nachweis zu führen, in dem Zugriffskontrollen für jede im Archiv gespeicherte elektronische Datendatei aufgeführt sind, und außerdem Mittel, um den Zugriff auf jeden Nachweis auf einen Computer mit Deponierungsrecht zu beschränken, Mittel, um eine Aufzeichnung zu führen, in der Zugriffsrechte auf die elektronischen Datendateien für jeden Computer mit Zugriffsrecht auf mindestens eine elektronische Datendatei im Datenarchiv aufgeführt sind, und Mittel zum Aktualisieren der Aufzeichnung für jeden Computer, der von einer Zugriffsrechtänderung in einem Nachweis betroffen ist.

In der Erfindung werden auch Datenträger bereitgestellt, die mit Programmcode zur Realisierung des oben beschriebenen Systems oder Verfahrens codiert sind.

Kurzbeschreibung der Zeichnungen

Im folgenden werden Ausführungsbeispiele der Erfindung ausführlich in Verbindung mit den beigefügten Zeichnungen beschrieben. Die Zeichnungen haben folgenden Inhalt:

Fig. 1 ist eine Schemazeichnung von einem Dokumentarchivsystem, das von einer dritten Partei verwaltet wird.

Fig. 2 ist eine Schemazeichnung, ähnlich wie **Fig.** 1, in der ein Tresor-Dokumentarchivsystem dargestellt ist, das in der bevorzugten Ausführungsform der vorliegenden Erfindung verwendet wird.

Fig. 3 ist ein Flußdiagramm des Dokumenterstellungsverfahrens gemäß der Erfindung.

Fig. 4, bestehend aus **Fig.** 4A und **Fig.** 4B ist ein Flußdiagramm des Dokumentabrufverfahrens gemäß der Erfindung.

Fig. 5A und 5B sind Flußdiagramme eines Verfahrens, gemäß der bevorzugten Ausführungsform der Erfindung, das für die Unveränderlichkeit der Zugriffskontrolle für Dokumentsuche und -abruf sorgt.

Fig. 6 schließlich ist ein Flußdiagramm eines erfindungsgemäßen Verfahrens zur Zuordnung von Eignerzugriffsrechten auf gespeicherte Dokumente.

Ausführliche Beschreibung der bevorzugten Ausführungsformen

Eine konventionelle Anordnung für ein Dokumentarchivsystem, bei dem eine dritte Partei als Verwalter agiert, ist in **Fig.** 1 dargestellt. Ein Dokumenturheber **100** kann Dokumente über seine Verbindung **102** mit einem fernen Dokumentarchivdienst **104**, z. B. einer von einer dritten Partei verwalteten Datenbank, deponieren. Als Eigner der deponierten Dokumente kann der Urheber **100** Zugriffsrechte auf die Dokumente zuweisen. Der Urheber eines Dokuments kann beispielsweise festlegen, dass ein Geschäftspartner **106** die "Lese"-Berechtigung hat, d. h. dass er das Dokument über seine Verbindung **108** mit dem Dokumentarchivdienst **104** abrufen, aber nicht ändern darf.

In solchen konventionellen Systemen ist das vom Urheber **100** deponierte Dokument normalerweise nicht verschlüsselt, so dass der Geschäftspartner **106** das Dokument auf Verlangen prüfen kann. Der Grund dafür ist, dass es nach dem Stand der Technik Probleme mit der Dechiffrierung von Dokumenten gibt. Für die Dechiffrierung eines Dokuments ist der Zugriff auf den privaten Schlüssel des Dokumenturhebers **100** erforderlich. Um den Zugriff auf seinen privaten Schlüssel zu ermöglichen, muß der Dokumenturheber **100** entweder selber zu allen Zeiten, zu denen möglicherweise eine Dechiffrierung angefordert werden könnte, online erreichbar sein, um die Dechiffrierung selber vorzunehmen (die Frage der Systemverfügbarkeit), oder er muß im voraus einen Plan entwickeln, um seinen privaten Schlüssel dem Geschäftspartner **106** direkt oder über einen vertrauenswürdigen Proxy-Server (nicht dargestellt) zukommen zu lassen.

In der US-Patentschrift Nr. 5,491,750 der International Business Machines Corporation, mit dem Titel "Method and Apparatus for Three-Party Entity Authentication and Key Distribution Using Message Authentication Codes", wird ein System beschrieben, das die Verteilung privater Sitzungsverwaltungsschlüssel ermöglicht, die von zwei oder mehr Kommunikationspartnern gemeinsam benutzt werden können, nachdem die Kommunikationspartner durch einen vertrauenswürdigen Vermittler authentifiziert worden sind. Die so erzeugten Schlüssel und andere ähnliche sind aber kurzlebig und ihre Verwendung sollte auf das absolut Notwendige beschränkt werden. Es ist nicht klar, dass ein solches Konzept geeignet wäre, Dechiffrierschlüssel in einem Dokumentrevisionssystem mit einem dauerhaften Dokumentarchiv sicher zwischen Kommunikationspartnern zu übertragen.

In konventionellen Systemen, in denen Dokumente für einige Zeit deponiert werden und nicht chiffriert sind (**Fig.** 1), muß darauf vertraut werden, dass die dritte Partei, die den Archivdienst **104** verwaltet, die Integrität des Dokuments bewahrt.

Das Dokumentarchivsystem in der bevorzugten Ausführungsform der vorliegenden Erfindung ist mit dem Produkt IBM Vault Registry erstellt, das Gegenstand der US-Patentanmeldung Nr. 980,022 mit dem Titel "Secure Server and Method of Operation for a Distributed Information System", eingereicht am 26. November 1977 und der IBM Corporation übertragen, ist. U. S. Die Patentschrift Nr. 980,022 ist hiermit durch Bezugnahme Teil des vorliegenden Dokuments. Das Produkt IBM Vault Registry bietet eine erweiterte Webserver-Umgebung, die eine sichere Erweiterung, einen sogenannten Tresor, der Klientenumgebung implementiert. Dieses System vertraut auf die im Hintergrund der Erfindung beschriebene moderne Übertragungstechnologie, dass die elektronische Übertragung von Dokumenten und anderen Daten intakt und fehlerfrei ankommt. Ressourcen in einem Client-Tresor sind nur verfügbar, wenn der Zugriff vom Client mit einer starken Authentifizierung mit Hilfe von zertifizierten öffentlichen Schlüsseln erfolgt. Abhängig von der Umgebung kann der Zugriff über den Web-Browser des Client erfolgen.

Der Informationsgehalt des Tresors ist aus Gründen der Vertraulichkeit chiffriert. Jeder Tresor auf einem Server besitzt einen eindeutigen Chiffrierschlüssel und Mechanismen, die den Zugriff auf die Schlüssel verhindern, sofern er nicht über den vom Eigner des Tresors genehmigten vertrauenswürdigen Pfad, z. B. einen Browser, erfolgt. Programme, die in einem Tresor laufen, sind durch Betriebssystemdienste isoliert, um folgendes zu gewährleisten:

a) dass sie in einem Prozeß mit einer Systemidentität (einem virtuellen Logon) laufen, so dass die Identität abhängigen Prozessen zur Verfügung steht, ohne dass eine Änderung durch ein im Tresor laufendes Programm möglich ist;
b) dass sie auf den Dateninhalt des Tresors, in dem sie laufen, zugreifen können – aber auf keinen anderen;
c) dass sie vom Eigner des Tresors für die Ausführung im Tresor genehmigt werden; und

d) dass sie signiert sind, um Manipulationen und Angriffe durch sog. "Trojanische Pferde" zu verhindern.

Programme, die in einem Tresor laufen, können Informationen in dem gleichen Tresor oder in anderen Tresoren, die gegenseitig sicheren Zugriff ihre öffentlichen Schlüssel haben, deponiert werden. Normalerweise befinden sich diese Tresore auf dem gleichen Tresorserver, sie können aber auch auf verschiedenen Tresorservern mit Zugriff auf eine gemeinsame Zertifizierungsstelle liegen, die die Information zum öffentlichen Schlüssel liefert. Im Zusammenhang mit einem Tresorarchiv kann "deponieren" verschiedenes bedeuten. In einer Implementierung kann "deponieren" die Chiffrierung der Daten im Chiffrierschlüssel des Zieltresors und die Signierung der Daten im Signierschlüssel des deponierenden Tresors bedeuten. Tresorprogramme können nicht direkt auf Chiffrier- oder Signierschlüssel zugreifen. Dies geschieht über eine API. Optional kann die "Deponierungs"-Funktion Informationen in eine Warteschlange im Zieltresor schreiben. Eine andere Option bietet einen "Deponierungsbeweis", der bestätigt, dass die Information deponiert wurde, und dass ein Programm im Zieltresor die Daten geöffnet hat. All diese "Deponierungs"-Funktionen bieten ein Mittel, um Informationen so zwischen Tresoren auszutauschen, dass:

a) ihr Ursprungsprozeß nicht geleugnet werden kann;
b) ihr Inhalt nicht von denen, die die Interprozeßkommunikationspuffer inspizieren, eingesehen werden kann; und
c) die Zustellung gewährleistet ist.

Wenn eine Anwendung keine Daten in die Warteschlange des Zieltresors stellen will, kann sie sich dafür entscheiden, die Information in einer Datei oder Datenbank zu speichern oder andere Systemdienste zu benutzen, die die Daten als "undurchsichtiges" Element behandeln können (z. B. Serialisierung für die Fortdauer des Objekts). Diese undurchsichtige Information kann mit Standardverfahren zum Zweck der Sicherung und Wiederherstellung verwaltet werden. Ihr Inhalt kann jedoch nur von einem im Kontext des Eignertresors laufenden Programm mit Hilfe der Sicherverwahrungs-Anwendungsprogrammschnittstelle dechiffriert werden. Mit dem Produkt IBM Vault Registry wurde die bevorzugte Ausführungsform der Erfindung entwickelt wie in **Fig.** 2 schematisch dargestellt.

Wie in dem System aus **Fig.** 1 kann auch in dem in **Fig.** 2 dargestellten Konzept ein Dokumenturheber **200** Dokumente über seine Verbindung **202** zu einem Dokumentarchivdienst **204** Dokumente deponieren und als Eigner der deponierten Dokumente dritten Parteien **206**, z. B. Geschäftspartnern, die über ihre eigenen Netzwerkverbindungen **208** auf die Dokumente im Dokumentarchivdienst **204** zugreifen können, Zugriffsrechte auf die Dokumente zuordnen. Anders als bei dem oben beschriebenen System sind die Benutzer des Dokumentarchivsystems aber nicht gezwungen, darauf zu vertrauen, dass die dritte Partei die Integrität der im Archiv hinterlegten Dokumente bewahrt.

Das Dokumentarchivsystem **204** in der bevorzugten Ausführungsform besteht aus zwei Komponenten, einem Anwendungsserver **210** und einem Tresor-Controller **214**. Der Anwendungsserver (AS) ist ein Programm zur Verwaltung des Datenbankarchivs **212**, das sich auf dem gleichen System oder auf einem fernen System in einem abgeschlossenen Netzwerk befindet. Der Tresor-Controller **214** enthält mehrere Komponenten: Benutzertresore **216**, **218**, die individuell den Dokumenturhebern **200** und Geschäftspartnern **206** zugeteilt sind, einen AS-Tresor **220**, der dem Anwendungsserver **210** zugeteilt ist, und ein Tresor-Überwachungsprogramm **222**.

Ein Benutzertresor **216** oder **218** ist nur für den Benutzer (Dokumenturheber **200** oder Geschäftspartner **206**) zugänglich, dem der Tresor zugeordnet ist, und nur nach ordnungsgemäßer Authentifikation. Die einzelnen Tresore haben keinen direkten Zugriff auf die Dokumentdatenbank **212**; der Zugriff erfolgt über den AS-Tresor **220** und den Anwendungsserver **210**.

Die Anwendungsserver-Komponente **210** läuft nicht auf einer vertrauenswürdigen Computerbasis, sondern kann auf jeder beliebigen Plattform ausgeführt werden. Der Anwendungsserver besitzt eine Gegenkomponente, die im AS-Tresor **220**, der ihm im Tresorserver **214** zugeteilt ist, läuft. Der AS-Tresor **220** kann mit dem Anwendungsserver **210** kommunizieren und hat über den Anwendungsserver Zugriff auf die Dokumentdatenbank **212**.

Fig. 3 ist ein Flußdiagramm des Dokumenterstellungsprozesses gemäß der bevorzugten Ausführungsform der Erfindung. In der Umgebung von IBM Vault Registry ist ein persönlicher Tresor im Prinzip eine sichere Erweiterung der Umgebung des Tresoreigners. Die Interaktion zwischen den Prozeßschritten in **Fig.** 3 ist deshalb zwischen den Tresoren des Dokumenturhebers und des Anwendungsservers dargestellt.

Wenn ein Dokument in dem Datenarchiv erstellt wird, wird es zuerst vom Arbeitsplatz des Benutzers, der es erstellt hat oder sein Urheber ist, in den persönlichen Tresor des Benutzers (Dokumenturhebers) gesendet (Block **300**), wo das Dokument mit dem privaten Signierschlüssel des Benutzertresors "signiert" wird (Block **302**).

Mit einer elektronischen Signatur eines Datenelementes garantiert der Signierende die Integrität des Datenelementes. Eine Signatur kann berechnet werden, indem zuerst ein Digest des Datenelementes berechnet wird. Das Digest ist eine relativ kleine Struktur (z. B. 128 Bit für eine MD2- oder MD5-Zusammenfassung) mit bestimmten Eigenschaften, um die Sicherheit zu gewährleisten. Erstens ist sie eine Einwegfunktion, d. h. aus einem Digest kann das Originaldokument, aus dem es hervorgegangen ist, nicht reproduziert werden. Außerdem ist es unmöglich (oder computertechnisch nicht machbar), zu einem Digest ein zweites Vor-Bild zu finden, das das gleiche Digest hat. Ferner ist das Digest auch kollisionsresistent. Das heißt, es ist äußerst unwahrscheinlich, dass zwei verschiedene Vor-Bilder das gleiche Digest erzeugen.

Das Digest des Datenelementes wird dann mit dem privaten Signierschlüssel der Benutzertresoranwendung chiffriert (Block **304**). In der bevorzugten Ausführungsform wird sowohl ein symmetrisches als auch ein asymmetrisches Kryptoghraphieverfahren mit öffentlichem Schlüssel benutzt.

Bei der Kryptographie mit öffentlichem Schlüssel besitzt eine Anwendung zwei Schlüssel, einen öffentlichen und einen privaten, die als Schlüsselpaar bezeichnet werden. Der private Schlüssel wird von der Anwendung lokal gespeichert und wird weiter unten ausführlicher beschrieben. Der öffentliche Schlüssel ist für alle Benutzer zugänglich, in der Regel über einen Verzeichnisdienst, z. B. X500. Die Verteilung öffentlicher Schlüssel ist in Fachkreisen bekannt und wird in der vorliegenden Spezifikation nicht weiter erläutert.

Wenn eine Kryptographie mit öffentlichem Schlüssel verwendet wird, kann ein mit dem öffentlichen Schlüssel chiffriertes Datenelement nur mit dem zugehörigen privaten Schlüssel dechiffriert werden. Entsprechend kann ein mit dem privaten Schlüssel chiffriertes Datenelement nur mit dem öffentlichen Schlüssel dechiffriert werden.

In einer Technologie mit symmetrischem Schlüssel wird für Chiffrierung und Dechiffrierung derselbe Schlüssel ver-

wendet. In der derzeitigen Praxis erfolgen Chiffrierung/Dechiffrierung und Schlüsselgenerierung bei der Technologie mit symmetrischem Schlüssel wesentlich schneller als bei der asymmetrischen Technologie mit öffentlichem Schlüssel.

Daten werden normalerweise mit einem nach dem Zufallsprinzip generierten symmetrischen Schlüssel chiffriert. Dann wird der symmetrische Schlüssel selber mit dem öffentlichen Chiffrierschlüssel des Benutzers chiffriert und mit dem Dokument gespeichert, so dass er Teil des Dokuments wird.

In **Fig.** 3 wird das chiffrierte Dokument und die elektronische Signatur zum Zweck der Aufbewahrung an den Tresor des Anwendungsservers gesendet (Block **306**). Nach Empfang des chiffrierten Dokuments (Block **308**) beglaubigt die im Tresor des Anwendungsservers laufende Anwendung die Signatur (Block **310**), indem sie mit ihrem eigenen privaten Signierschlüssel noch einmal signiert.

Die Beglaubigung einer Signatur in einem elektronischen Kontext bedeutet, dass eine dritte Partei, die als "Notar" fungiert, den Inhalt einer Signatur zertifiziert. (Die Begriffe "Notar" und "beglaubigen" haben in dieser Spezifikation nicht den vollen Bedeutungsumfang aller Pflichten die einem Notariat von einer Regierungsbehörde übertragen werden.) Allgemein erfolgt eine elektronische Beglaubigung einer Signatur als zusätzliche Vorsichtsmaßnahme, um eine spätere unberechtigte Änderung der Signatur zu verhindern. Im Fall der vorliegenden Erfindung verhindert die Beglaubigung einer digitalen Signatur des Benutzers, dass dieser das Originaldokument im Dokumentarchiv ersetzt oder ändert. Eine Prüfung der beglaubigten Signatur des Dokuments würde jegliche Inkonsistenz ans Tageslicht bringen.

Eine beglaubigte elektronische Signatur enthält zwei Informationen, nämlich die Signatur des betreffenden Datenelements durch den Urheber und die Signatur der Urhebersignatur durch den Notar. Die Signatur des Notars sollte über die Urhebersignatur und den aktuellen Zeitstempel berechnet werden.

Die Anwendung, die im Tresor des Anwendungsservers läuft, signiert dann das von ihr empfangene Dokument (Block **312**). Da die Daten, die er vom Dokumenturheber empfängt, chiffriert sind, kennt der Anwendungsserver faktisch den Inhalt des Dokuments nicht. Deshalb wird gemäß der Erfindung diese zweite Signatur über das chiffrierte Dokument und die beglaubigte Urhebersignatur berechnet. Die Signatur des Anwendungsservers stellt einen Empfangsbeweis dar, der dem Dokumenturheber (demjenigen, der das Dokument deponiert), beweist, dass der Archivdienst das Dokument empfangen hat. Die Erstellung des Dokuments im Archiv kann dann später nicht mehr vom Archivdienst geleugnet werden.

Das chiffrierte Dokument, die beglaubigte Urhebersignatur und der Empfangsbeweis werden im Archiv des Anwendungsservers oder in der Anwendungsdatenbank gespeichert (Block **314**). Der Empfangsbeweis wird an den Tresor des Dokumenturhebers gesendet (Block **316**). Der Tresor des Dokumenturhebers prüft die Richtigkeit des Empfangsbeweises (Block **318**), indem die Signatur des chiffrierten Dokuments überprüft wird. Der Tresor des Dokumenturhebers prüft auch die Aktualität des Zeitstempels in der beglaubigten Signatur (Block **320**). Die Toleranz für den Zeitstempel hängt von der Anwendung ab. Wenn bei einer dieser Prüfungen ein Fehler erkannt wird, wird eine Fehlermeldung an den AS-Tresor gesendet (Block **322**) und im System protokolliert. Wenn der Empfang korrekt und aktuell ist, sendet die Anwendung, die im Tresor des Benutzers läuft, den Empfangsbeweis an den verursachenden Benutzer zurück (Block **324**), damit sie für eine spätere Referenz in einem lokalen Cache gespeichert wird, falls bewiesen werden muß, dass das Dokument im Archiv gespeichert worden ist.

Es ist möglich, dass der Dokumenturheber das Dokument mit einem eigenen Verfahren signieren und/oder chiffrieren kann, bevor er es zur Speicherung in seinen Tresor sendet. Das Dokumentarchiv beachtet den Inhalt des zu speichernden Dokuments aber nicht. Ein chiffriertes Dokument wird deshalb vom Tresor des Benutzers erneut signiert und chiffriert, wie jedes andere Dokument.

Fig. 4 ist ein Flußdiagramm, in dem dargestellt ist, welche Schritte gemäß der bevorzugten Ausführungsform der Erfindung ausgeführt werden müssen, damit das Dokument von einem anfordernden Benutzer abgerufen werden kann, der unter einem von Dokumenturheber verwalteten Nachweistyp, der als Zugriffskontroll-Liste (ACL) bezeichnet wird, autorisiert worden ist. Wie in **Fig.** 3 sind die Verfahrensschritte zwischen drei Aktoren, nämlich Benutzer, Anwendungsserver und Anforderer, aufgeteilt, auf der Basis, dass deren persönliche Tresore im Prinzip sichere Erweiterungen ihrer betreffenden Arbeitsbereiche sind.

In **Fig.** 4A stellt der Benutzer an seine Tresoranwendung eine Anforderung, ein Dokument aus dem Anwendungsserverarchiv abzurufen (Block **400**), und seine Tresoranwendung sendet dann ihrerseits die Dokumentabrufanforderung an den Anwendungsserver-Tresor (Block **402**).

Die Tresoranwendung des Anwendungsservers empfängt die Zugriffsanforderung (Block **404**) und ruft das chiffrierte Dokument und die beglaubigte Signatur aus der Anwendungsdatenbank ab.

Die Tresoranwendung des Anwendungsservers sendet das chiffrierte Dokument und die beglaubigte Signatur an den Tresor des Dokumenturhebers. Der Tresor des Anwendungsservers sendet auch die Identität des anfordernden Benutzertresors an den Tresor des Urhebers (Block **408**).

Der Tresor des Urhebers prüft, ob der anfordernde Benutzer die Berechtigung zum Abrufen des Dokuments besitzt (Block **410**). In der bevorzugten Ausführungsform wird die Dokumentzugriffskontrolle durch Zugriffskontroll-Listen aktiviert, mit denen der Zugriff auf das Dokument auf autorisierte Stellen beschränkt wird. Eine Zugriffskontroll-Liste (ACL) ist einem Dokument zugeordnet und wird im Tresor des Dokumenturhebers gespeichert und verwaltet wie weiter unten im Zusammenhang mit **Fig.** 5A und **Fig.** 6 beschrieben. Die ACL muß geprüft werden, wenn ein Benutzer eine Dokumentabrufanforderung sendet. Ein anfordernder Benutzer erhält nur eine Kopie des Dokuments, wenn er das Zugriffsrecht besitzt.

In der bevorzugten Ausführungsform der Erfindung können Fähigkeitslisten benutzt werden, damit anfordernde Benutzer ihr Zugriffsrecht auf Dokumente im voraus verifizieren können. In einer Fähigkeitsliste sind alle Dokumente in einem Archiv aufgeführt, für die ein bestimmter Benutzer das Zugriffsrecht besitzt. Die Fähigkeitsliste eines anfordernden Benutzers wird in seinem eigenen Tresor gespeichert und verwaltet. Der Anfordernde braucht nur diese Liste durchzusehen, um festzustellen, auf welche Dokumente er zugreifen kann. Verwendung und Verwaltung der Fähigkeitslisten werden im Zusammenhang mit **Fig.** 5B ausführlicher beschrieben.

Wenn der anfordernde Benutzer keine Zugriffsberechtigung auf das Dokument besitzt, wird eine Fehlermeldung an den Urheber gesendet und im System protokolliert (Block **414**).

In **Fig.** 4B wird, wenn der anfordernde Benutzer die Zugriffsberechtigung für das Dokument besitzt, das Dokument von der Tresoranwendung des Urhebers dechiffriert (Block **416**) und die beglaubigte Signatur überprüft (Block **418**). Da

die Originalsignatur des Urhebers über den unchiffrierten Dokumentinhalt berechnet wurde, können nur diejenigen Benutzer, die auf den Dokumentinhalt zugreifen können (d. h. die den privaten Schlüssel des Urhebers besitzen), die Signatur prüfen. Wenn die empfangene Signatur nicht dem entspricht, was der Dokumenturheber in seinen eigenen Dateien stehen hat, ist klar, dass es sich nicht um dieselbe Version des Dokuments handelt, die deponiert wurde, und der Urheber sendet eine Fehlernachricht an den Anwendungsserver (Block **420**).

Wenn die Signatur geprüft worden ist, sendet der Urheber das dechiffrierte Dokument und die beglaubigte Signatur an den Tresor des anfordernden Benutzers (Block **422**).

Nach Empfang des dechiffrierten Dokuments versucht die Tresoranwendung des anfordernden Benutzers, die beglaubigte Signatur des Urhebers zu prüfen (Block **424**). Wenn der anfordernde Benutzer sie nicht verifizieren kann, wird eine Fehlermeldung an den Urheber gesendet und im System protokolliert (Block **426**).

Wenn die beglaubigte Signatur des Urhebers verifiziert werden kann, signiert der Tresor des Anfordernden die mit dem Dokument empfangene beglaubigte Signatur. Diese Signatur wird über die beglaubigte Signatur und über den aktuellen Zeitstempel berechnet und stellt einen Zustellungsbeweis dar (Block **428**), der belegt, dass der anfordernde Benutzer das Dokument aus dem Archiv abgerufen hat. Der Tresor des Anfordernden sendet das dechiffrierte Dokument zusammen mit dem von ihm generierten Empfangsbeweis an den Arbeitsplatz des Anfordernden (Block **430**). Der Tresor des Anfordernden sendet auch den Empfangsbeweis an den Anwendungsserver-Tresor (Block **432**). Der Anwendungsserver verifiziert die Signatur des Anforderertresors auf dem Empfangsbeweis (Block **434**). Wenn die Signatur nicht verifiziert werden kann, wird eine Fehlermeldung an den Urheber gesendet und im System protokolliert (Block **436**). Wenn die Signatur verifiziert werden kann, speichert der Anwendungsservertresor den Beweis in den Anwendungsdatenbank, falls der Anwendungsserverspäter nachweisen muß, dass der Anfordernde das Dokument tatsächlich abgerufen hat.

Unveränderlichkeit der Zugriffskontrolle für den Dokumentabruf

Wie bereits erwähnt besteht in einem Datenarchiv die Notwendigkeit eine Dokumentzugriffskontrolle. Dies bedeutet, dass nur die vom Dokumenteigner autorisierten Benutzer Einsicht in die Dokumente haben, und dass Dokumentzugriffserlaubnisse nur vom Dokumenteigner (d. h. vom Urheber) selber und von den Personen, die vom Dokumenteigner die Berechtigung zum Ändern der Zugriffskontroll-Liste für das Dokument erhalten haben, geändert werden können. Es ist wichtig, dass sichergestellt ist, dass selbst der Archivverwalter nicht in der Lage ist, ohne Autorisierung durch den Dokumenteigner die Zugriffsberechtigungen für ein Dokument zu ändern.

Es gibt zwei verschiedene Arten von Anwendungsanforderungen für die Unveränderlichkeit der Dokumentzugriffskontrolle. Der Dokumentzugriff muß in folgenden Fällen geprüft werden:

1) wenn ein Benutzer eine Suche durchführt, um alle Dokumente zu finden, für die er die Berechtigung zum Betrachten hat und
2) wenn ein Benutzer tatsächlich ein Dokument abruft.

Alle Anwendungen müssen die Zugriffskontrolle beim Dokumentabruf (Zugriffsart 2) erzwingen. Für diese Zugriffsart muß das Archiv garantieren, dass die Zugriffskontrolle eines Dokuments nicht von einem nicht autorisierten Benutzer, z. B. einem Konkurrenten, geändert werden kann.

In einigen Anwendungen ist es aber nicht erforderlich, dass ein Benutzer nicht das Dokument abfragen kann, um festzustellen, welche Dokumente er betrachten darf. Dieses Wissen kann z. B. offline in geschäftlichen Besprechungen oder telefonisch übermittelt werden. In einem solchen Fall weiß der Benutzer bereits, auf welche Dokumente er zugreifen kann, und seine Kenntnis seines eigenen Dokumentzugriffs kann nicht von Aktionen des Archivs beeinflußt werden.

Ein System, das die Unveränderlichkeit der Zugriffskontrolle nur beim Dokumentabruf, aber nicht bei der Dokumentsuche erzwingt, ist Gegenstand unserer gleichzeitigen Anmeldung mit dem Titel "System for Electronic Repository of Data Enforcing Access Control on Data Retrieval" (kanadische Patentanmeldung 2,256,934). Die diesem System wird die Zugriffskontrollinformation in der Datenbank bzw. im Archiv des Anwendungsservers gespeichert.

Eine strengere Form der Unveränderlichkeit der Zugriffskontrolle, die dort verwendet werden sollte, wo Benutzer nicht über unabhängige Information über ihren Dokumentzugriff verfügen, betrifft sowohl die Dokumentsuche als auch den Dokumentabruf. Für diese Forderung kann die Zugriffskontrollinformation nicht in der Anwendungsdatenbank gespeichert werden. Statt dessen wird sie im Tresor des Dokumenteigners gespeichert. Dieses Schema ist Gegenstand der vorliegenden Erfindung und wird durch die Flußdiagramme in **Fig.** 5 und **Fig.** 6 illustriert und weiter unten beschrieben.

In der vorliegenden Ausführungsform ist jedem Dokument eine Zugriffskontroll-Liste (ACL) zugeordnet, die die Dokumentzugriffsberechtigung verschiedener Benutzer festlegt. Außerdem besitzt jeder Benutzer im System eine Fähigkeitsliste, in der alle gespeicherten Dokumente, von denen der Benutzer nicht der Eigner ist, auf die er aber zugreifen kann, identifiziert werden.

Um die Unveränderlichkeit zu garantieren, wird jede ACL im Tresor des Dokumenturhebers verarbeitet, wie in **Fig.** 5A dargestellt, und parallel dazu wird jede Fähigkeitsliste im entsprechenden Benutzertresor verarbeitet wie in **Fig.** 5B dargestellt.

In **Fig.** 5A stellt der Tresor des Dokumenteigners nach einer Aktualisierung einer ACL (Block **500**) fest, welche Benutzer von der Änderung betroffen sind (Block **502**), und eine Nachricht, in der die Art der Zugriffsänderung (Hinzufügung, Erweiterung oder Beschränkung) angegeben wird, wird im Tresor jedes Benutzers deponiert, dessen Zugriffsrecht auf das Dokument geändert worden ist (Block **504**).

Jeder ACL ist eine Versionsnummer und ein Zeitstempel der letzten Änderung zugeordnet. Der Tresor des Dokumenteigners erhöht dann inkrementell die Versionsnummer der ACL (Block **506**) und ersetzt deren alten Zeitstempel durch den aktuellen Zeitstempel (Block **508**). Aus der aktuellen Versionsnummer und dem Zeitstempel, die der ACL jetzt zugeordnet sind, wird ein Token, das die Unveränderlichkeit der ACL garantieren soll, erstellt und vom Tresor des Dokumenturhebers signiert (Block **510**). Die ACL wird ebenfalls vom Tresor des Dokumenturhebers signiert (Block **512**).

Das ACL-Token wird dann an den Tresor jedes zum Zugriff auf das Dokument berechtigten Benutzers gesendet, wo es zur Speicherung mit der Zugriffsanwendung des Benutzers auf dessen Arbeitsplatz gespeichert wird (Block **514**), damit eine spätere Verifizierung der ACL möglich ist. Das signierte Token wird zur Speicherung an den Arbeitsplatz des Dokumenturhebers gesendet (Block **516**). Da der Doku-

menturheber eine Kopie des signierten Tokens besitzt, wird er letztlich zum Arbiter darüber, ob die Dokument-ACL aktuell ist oder nicht.

Wenn ein Geschäftspartner ein Dokument abrufen möchte, sendet die AS-Tresoranwendung das chiffrierte Dokument wie oben beschrieben an den Tresor des Urhebers (Block **408** in **Fig.** 4A). Um die Berechtigung des Anfordernden zu verifizieren (Block **412** in **Fig.** 4A), schaut der Tresor des Dokumenturhebers einfach in der lokal gespeicherten verifizierten ACL nach, ob der Anfordernde das Zugriffsrecht auf das angegebene Dokument besitzt. Mit diesem Verfahren kann niemand die in der Anwendungsdatenbank gespeicherte ACL ändern, ohne dass dies vom Tresor des Dokumenturhebers bemerkt wird.

Wie oben beschrieben besitzt jeder Benutzer, der Eigner von Dokumenten im Archiv ist, auf seinem Arbeitsplatz die signierten Tokens der korrekten Version jeder ACL. Die ACL-Versionen im Benutzertresor werden verifiziert, indem das auf dem Arbeitsplatz des Benutzers gespeicherte Token mit dem im Benutzertresor gespeicherten verglichen wird. Dieser Vergleich kann zu verschiedenen Zeiten ausgeführt werden; eine gute Gelegenheit zur Verifizierung der in einem Benutzertresor gespeicherten ACLs ist das Logon, so dass jedesmal, wenn sich ein Benutzer beim System anmeldet, die ACLs verifiziert werden.

Wenn die Verifizierung der ACL nicht gelingt, kann die Benutzertresoranwendung automatisch die Verarbeitung jeglicher Anforderung, ein von der ACL geschütztes Dokument abzurufen, einstellen. Dieser Zustand der Unveränderlichkeit des Dokuments würde weiterbestehen, bis der Benutzer entweder eine neue ACL erstellt oder die vorhandene ACL neu zertifiziert. Der Prozeß der Rezertifizierung der vorhandenen ACL würde die Synchronisierung des im Benutzertresor gespeicherten ACL-Tokens mit dem auf dem Arbeitsplatz des Benutzers gespeicherten Token einschließen.

Bei jeder Aktualisierung einer ACL werden parallel zu den in **Fig.** 5 A aufgeführten Schritten einige andere Schritte ausgeführt. Diese zusätzlichen Schritte sind in **Fig.** 5B dargestellt.

Jeder Benutzertresor ist für die Verwaltung einer Fähigkeitsliste zuständig, die eine Auflistung aller Dokumente, auf die der Benutzer zugreifen darf, enthält. Die Aktualität der Fähigkeitsliste selber wird durch eine Versionsnummer und einen neuesten Zeitstempel identifiziert. Wenn eine Nachricht, die eine Änderung der Zugriffsmöglichkeit eines Benutzers auf ein Dokument (eine Aktualisierung einer Dokument-ACL) mitteilt, im Tresor des Benutzers eingeht (Block **520**), wird die Fähigkeitsliste im Tresor des Benutzers automatisch mit Versionsnummer (Block **522**) und neuestem Zeitstempel (Block **524**) aktualisiert. Über die Versionsnummer und den Zeitstempel (Block **526**) wird ein Token berechnet, das zur Verifizierung der Richtigkeit der Fähigkeitsliste verwendet werden kann. Das Token wird vom Tresor des Benutzers signiert (Block **528**), und die Fähigkeitsliste ebenfalls (Block **530**). Das signierte Token und die signierte Fähigkeitsliste werden im Tresor des Benutzers gespeichert (Block **532**), der Tresor des Benutzers bewahrt aber die alte Fähigkeitsliste und ihr Token auf, da das Token für die alte Fähigkeitsliste dem auf dem Arbeitsplatz des Benutzers gespeicherten Token entspricht, bis eine Aktualisierung vorgenommen werden kann.

Eine Möglichkeit zur Synchronisation der aktuellen Fähigkeitsliste mit dem auf dem Arbeitsplatz gespeicherten Token des entsprechenden Benutzers besteht darin, dies automatisch zu tun, wenn sich der Benutzer beim System anmeldet (Block **532**). Die Richtigkeit des Tokens auf dem Arbeitsplatz des Benutzers kann mit dem im Tresor des Benutzers aufbewahrten alten Token verglichen werden, und dann kann das aktualisierte Token an den Arbeitsplatz des Benutzers gesendet werden (Block **534**). Sobald das alte Token auf dem Arbeitsplatz des Benutzers ersetzt worden ist, kann die alte Fähigkeitsliste und ihr Token aus dem Tresor des Benutzers gelöscht werden.

Eine andere Alternative zur Aktualisierung des Tokens der Fähigkeitsliste auf dem Arbeitsplatz des Benutzers (nicht dargestellt) zu aktualisieren, wäre, dass der Benutzer die Initative ergreifen muß, um Aktualisierungen der Fähigkeitsliste seit seiner letzten Anmeldung beim System festzustellen.

Um die Zusammengehörigkeit von ACLs und den Fähigkeitslisten sicherzustellen, muß die Umgebung, auf der das System basiert (z. B. das Produkt IBM Vault Registry) eine garantierte Nachrichtenzustellung für Nachrichten, die von einem Tresor in einem anderen deponiert werden, bieten. Die Garantie der Zustellung einer Fähigkeitsliste kann auch durch die Anwendung erfolgen, indem z. B. eine Bestätigung von dem Benutzer, der die Aktualisierung empfängt, gefordert wird.

Als Resultat dieses Schemas werden ACL und Fähigkeitsliste von ihren Eignern gespeichert. Keine Partei im System kann die Zugriffskontroll-Liste eines Dokuments ändern, ohne dass der Dokumenteigner dies erfährt. Außerdem kann keine Partei im System das Wissen eines Benutzers über sein Zugriffsrecht auf ein Dokument (d. h. eine Fähigkeit) ändern, ohne dass der autorisierte Benutzer dies bemerkt.

Im Gegensatz zu dem Zugriffskontrollschema, das in unserer oben genannten, gleichzeitig anhängigen Anmeldung beschrieben wird, wo die Suche im Tresor des Anwendungsservers stattfindet, erfolgt in der vorliegenden Erfindung die Suche nach Dokumenten, für die ein Benutzer die Zugriffsberechtigung besitzt, in der Tresoranwendung des Benutzers selber.

Zuordnung von Eignerzugriffsrechten

In manchen Umgebungen muß der Dokumenteigner die Möglichkeit haben, einer anderen Person die Erlaubnis zum Ändern der Zugriffsliste des Dokuments zu erteilen. Zum Beispiel wenn der Eigner nicht da ist, kann ein anderer autorisierter Benutzer in der Lage sein, die Zugriffskontrolle für das bestimmte Dokument zu aktualisieren.

In einer bevorzugten Ausführungsform der Erfindung kann die Aktualisierung von ACLs oder Fähigkeitslisten von anderen Benutzern im System durchgeführt werden, indem die in **Fig.** 6 dargestellten Schritte ausgeführt werden.

Zum Beispiel wenn eine Aktualisierung der ACL versucht wird, muß der Benutzer, der die Aktualisierung vornimmt, in der Lage sein, das aktuelle signierte Token für die ACL vorzulegen (Block **600**). Das signierte Token wird zum Tresor des Benutzers gesendet (Block **602**), der das signierte Token an den Tresor des Urhebers übergibt (Block **604**). Wenn dem aktualisierenden Benutzer in der ACL dieses Dokuments keine Eignerzugriffsrechte zugewiesen worden sind, dann erkennt der Tresor des Dokumenteigners dies, und er verweigert die Aktualisierung und sendet eine Fehlermeldung an den Tresor des Benutzers (Blöcke **606** und **608**).

Wenn der Tresor des Urhebers das Zugriffsrecht des signierenden Benutzers auf das Dokument verifizieren kann, und wenn festgestellt wird, dass die Versionsnummer und der Zeitstempel des ACL-Tokens aktuell sind (Block **606**), wird die ACL aktualisiert (Block **610**), und ein neues Token wird generiert und signiert (Block **612**) und im Tresor des Urhebers gespeichert (Block **714**). Das neu signierte Token

wird an den Tresor des Dokumenturhebers gesendet (Block **616**). Der Tresor des Aktualisierenden sendet das neue Token zur Speicherung an dessen Arbeitsplatz zurück (Block **618**). Das neu signierte Token kann optional auch zur Speicherung im Archiv an den Tresor des Anwendungsservers gesendet werden (Block **620**).

Dieses Verfahren verlangt, dass zu jedem Zeitpunkt nur eine einzige Person eine ACL-Aktualisierung durchführt. Wenn zum Beispiel ein Dokumenteigner John Urlaub nimmt, kann er einer Mitarbeiterin Mary erlauben, die ACL seines Dokuments in seiner Abwesenheit zu aktualisieren, indem er Mary sein aktuelles Token für die ACL des Dokuments gibt. Mary führt dann eine ACL-Aktualisierung durch, indem sie das Token durch ihren Tresor John's Tresor vorlegt. Mary empfängt das neu signierte Token für die ACL und gibt es John bei seiner Rückkehr wieder zurück. Nach der Installation des neuen Tokens kann John selber eine ACL-Aktualisierung vornehmen.

Datensicherung und -wiederherstellung

Gelegentlich kann es notwendig sein, dass der Verwalter des Dokumentarchivs die Dokumentdatenbank aus einem vorherigen Backup wiederherstellt. Dies kann beispielsweise bei einem katastrophalen Datenbankfehler, z. B. bei einem Festplattendefekt, der Fall sein.

Die zu sichernden Daten sind die Dokumente selber, die ACLs (entweder in der Anwenderdatenbank oder in den Eignertresoren gespeichert), die Fähigkeitenlisten (für die Systeme, in denen sie implementiert sind, wie oben beschrieben), und die Verifikationstokens von ACLs und Fähigkeitslisten.

Nach einer Rückspeicherung der Daten können Aktualierungen, die nach der letzten Sicherung vorgenommen wurden, verloren gegangen sein. Für die Zwecke der vorliegenden Erfindung könnte es sich dabei auch um ACL- und Fähigkeitslisten-Aktualisierungen handeln. Wenn dies geschieht, stimmen die auf den Benutzerarbeitsplätzen gespeicherten Verifizierungstokens möglicherweise nicht mehr mit den Tokens in den entsprechenden Tresoren überein, so dass die Benutzer keinen Zugriff mehr haben. Deshalb wurde als Standard für die Datenwiederherstellung in verschiedenen Situationen das folgende System implementiert. Es wird angenommen, dass die Sicherung zum Zeitpunkt ZEIT1 erfolgte, und die Rückspeicherung zu einem späteren Zeitpunkt ZEIT2. Wenn eine vollständige Rückspeicherung der Dokumentdatenbank, der ACLs, der Fähigkeitslisten und der entsprechenden in den Tresoren gespeicherten Tokens durchgeführt wird, können die Benutzer, die vor ZEIT1 auf ein Dokument zugreifen konnten, dies auch nach ZEIT2 tun. Dies bedeutet, dass wenn ein Benutzer vor ZEIT1 berechtigt war, die Berechtigung aber zwischen ZEIT1 und ZEIT2 widerrufen wurde, dieser Benutzer dennoch auf das Dokument zugreifen kann, bis der Eigner des Dokuments das ACL-Token prüft. Nach einer vollständigen Datenrückspeicherung sollten deshalb alle Benutzer eine Prüfung der ACL und der Fähigkeitsliste durchführen.

Wenn nur die Dokumentdatenbank zurückgespeichert wurde und die ACLs, die Fähigkeitslisten und die in den Tresoren gespeicherten Tokens unberührt geblieben sind, können Benutzer feststellen, dass sie das Zugriffsrecht für ein Dokument besitzen, das gar nicht in der Datenbank gespeichert ist, da das Dokument nach ZEIT1 hinzugefügt wurde, aber nachher bei der Rückspeicherung der Datenbank verloren gegangen ist. Da alle Tokens aktuell sind, gibt es keine weiteren Anomalien.

Ein anderer Fall liegt vor, wenn in einem System keine Fähigkeitslisten benutzt werden, die ACLs aber in der Anwendungsdatenbank gespeichert werden. Wenn die Dokumentdatenbank und die ACL zurückgespeichert worden sind, während die in den Tresoren gespeicherten Tokens nicht zurückgespeichert wurden, stellen die Benutzer fest, dass alle Dokumente, deren ACL nach ZEIT1 geändert wurden, nicht mehr zugänglich sind. Dies kommt daher, dass die ACL-Tokens in der Anwendungsdatenbank nicht mit den in den Tresoren der einzelnen Eigner gespeicherten Tokens übereinstimmen. Um dieses Problem zu lösen, müssen alle Dokumenteigner die ACLs aktualisieren. Eine Möglichkeit dazu ist, dass der Verwalter die alten ACLs (die zu ZEIT1 in Kraft waren), den Dokumenteignern sendet und sie bittet, die entsprechenden Tokens in ihren Tresoren neu zu installieren. Diese Aktualisierung wird manuell, nicht automatisch, vorgenommen, und die Dokumente eines Eigners sind unzugänglich, bis er die Aktualisierung durchgeführt hat.

In Situationen, in denen Datenbankinkonsistenzen vermieden werden müssen, kann der Archivverwalter nach einer Rückspeicherung den Zugriff auf alle Dokumente sperren, bis der Urheber Fehlerbehebungsmaßnahmen ergriffen hat. Diese Sperre kann für alle Dokumente im Archiv gelten oder nur für einen Teil der Dokumente, bei denen die Konsistenz am kritischsten ist. In diesem Fall muß man sich auf den Archivverwalter verlassen, um die Konsistenz des Systems zu wahren. Wie bereits erwähnt hat der Verwalter aber in keinem Fall die Möglichkeit, Benutzerzugriffsrechte auf ein Dokument zu erteilen oder zu widerrufen.

In der obigen Beschreibung wurden bevorzugte Ausführungsformen der vorliegenden Erfindung mittels des Produkts IBM Vault Registry beschrieben. Dem Fachmann ist aber klar, dass die vorliegende Erfindung auch mit anderen Produkten, die über ähnliche Funktionen verfügen, implementiert werden könnte, z. B. mit sicheren tresorähnlichen Umgebungen, die sich lokal auf dem Arbeitsplatz der einzelnen Benutzer befinden. Solche und andere Abwandlungen, die für den Fachmann offensichtlich sind, sollen ebenfalls unter den Schutzumfang der beigefügten Ansprüche fallen.

Patentansprüche

1. Ein sicheres System zum Suchen von elektronischen Datendateien, die in einem Datenarchivsystem gespeichert sind, umfassend:

eine Kommunikationsumgebung mit

(i) einem ersten Agentenprogramm für einen Computer, der eine elektronische Datendatei im Datenarchivsystem deponiert, und

(ii) einem zweiten Agentenprogramm für einen ersten Benutzercomputer mit Zugriffsrecht auf die elektronische Datendatei;

einer Nachweisliste für die elektronische Datendatei, in der Zugriffskontrollen für die elektronische Datendatei aufgeführt sind, wobei die Nachweisliste für das erste Agentenprogramm zugänglich ist und von diesem verwaltet wird;

eine erste Aufzeichnung der Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei, wobei die erste Aufzeichnung für das zweite Agentenprogramm zugänglich ist und von diesem verwaltet wird;

Mittel, um Änderungen an der Nachweisliste, die die Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei betreffen, vom ersten Agentenprogramm an das zweite Agentenprogramm zu senden, um die erste Aufzeichnung zu aktualisieren; und

Mittel, mit denen das erste Agentenprogramm die Zugriffsrechte des ersten Benutzercomputers auf die elek-

tronische Datendatei prüfen kann, bevor die elektronische Datendatei für das zweite Agentenprogramm freigegeben wird.

2. Das sichere System nach Anspruch 1, wobei das erste Agentenprogramm eine sichere Erweiterung des deponierenden Computers und das zweite Agentenprogramm eine sichere Erweiterung des ersten Benutzercomputers ist.

3. Das sichere System nach Anspruch 2, das außerdem Mittel besitzt, um die Änderungen der Nachweisliste, die die Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei betreffen, vom zweiten Agentenprogramm an den ersten Benutzercomputer zu senden.

4. Das sichere System nach Anspruch 1 oder 2, das außerdem folgendes umfaßt:

ein drittes Agentenprogramm für einen zweiten Benutzercomputer mit Zugriffsrecht auf die elektronische Datendatei; und

eine zweite Aufzeichnung der Zugriffsrechte des zweiten Benutzercomputers auf die elektronische Datendatei, wobei die Aufzeichnung für das dritte Agentenprogramm zugänglich ist und von diesem verwaltet wird,

und wobei das Mittel um die Änderungen an der Nachweisliste, die die Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei betreffen, zur Aktualisierung der ersten Aufzeichnung an das zweite Agentenprogramm zu übertragen, Mittel umfaßt, um Änderungen an der Nachweisliste, die die Zugriffsrechte des zweiten Benutzercomputers auf die elektronische Datendatei betreffen, zum Aktualisieren der zweiten Aufzeichnung vom ersten Agentenprogramm an das dritte Agentenprogramm zu übertragen; und

wobei das Mittel, mit dem das erste Agentenprogramm die Zugriffsrechte des ersten Benutzercomputers auf die elektronische Datendatei verifiziert, bevor die elektronische Datendatei für das zweite Agentenprogramm freigegeben wird, ein Mittel umfaßt, mit dem das erste Agentenprogramm die Zugriffsrechte des zweiten Benutzercomputers auf die elektronische Datendatei verifiziert, bevor die elektronische Datendatei für das dritte Agentenprogramm freigegeben wird.

5. Das sichere System nach Anspruch 4, wobei das dritte Agentenprogramm eine sichere Erweiterung des zweiten Benutzercomputers ist.

6. Das sichere System nach Anspruch 5, das außerdem Mittel besitzt, um die Änderungen der Nachweisliste, die die Zugriffsrechte des zweiten Benutzercomputers auf die elektronische Datendatei betreffen, vom dritten Agentenprogramm an den zweiten Benutzercomputer zu senden.

7. Das sichere System nach Anspruch 2 oder 5, wobei die Kommunikationsumgebung einen Server umfaßt.

8. Das sichere System nach Anspruch 1, 2 oder 5, das außerdem in der Kommunikationsumgebung eine Schnittstelle zum Datenarchivsystem umfaßt, die daran angepaßt ist, alle Übertragungen zwischen dem Datenarchivsystem und dem Agentenprogramm zu empfangen.

9. Das sichere System nach Anspruch 8, wobei die Schnittstelle eine sichere Erweiterung des Datenarchivsystems ist.

10. Ein Verfahren für die Verwaltung eines sicheren elektronischen Datensuchsystems für ein elektronisches Datenarchiv, wobei das System eine Nachweisliste, in der Zugriffsrechte auf die elektronische Datendatei im Datenarchiv aufgeführt sind, und eine Aufzeichnung, in der Dokumentzugriffsrechte für jeden Computer mit Zugriff auf die im Archiv gespeicherten elektronischen Daten aufgeführt sind, besitzt, wobei das Verfahren folgende Schritte umfaßt:

Aktualisieren einer Nachweisliste für eine im Archiv gespeicherte elektronische Datendatei;

Identifizieren aller Computer, deren Zugriffsrecht auf die elektronische Datendatei von der Aktualisierung betroffen ist;

Übertragen der Zugriffsänderung an alle betroffenen Computer;

Aktualisieren der Zugriffsrechtaufzeichnungen aller betroffenen Computer; und

Übertragen der aktualisierten Zugriffsrechtaufzeichnungen an die betroffenen Computer.

11. Ein sicheres System zum Suchen von elektronischen Datendateien, die in einem Datenarchivsystem gespeichert sind, umfassend:

Mittel zum Verwalten eine Nachweisliste, in der Zugriffskontrollen für jede im Datenarchivsystem gespeicherte elektronische Datei aufgeführt sind;

Mittel zum Beschränken des Zugriffs auf jede Nachweisliste auf einen Computer mit Deponierungsberechtigung;

Mittel zum Verwalten einer Aufzeichnung, in der die Zugriffsrechte auf die elektronische Datendatei für jeden Computer mit Zugriffsrecht auf mindestens eine elektronische Datendatei im Datenarchivsystem aufgeführt sind;

Mittel, um den Zugriff auf die Aufzeichnung auf den zugehörigen Computer mit Zugriffsrechten zu beschränken; und

Mittel zum Aktualisieren der Aufzeichnung für jeden Computer, der von einer Zugriffsänderung in einer Nachweisliste betroffen ist.

12. Ein computerlesbarer Speicher zum Speichern der Instruktionen zur Verwendung bei der Ausführung des Verfahrens nach Anspruch 10 auf einem Computer.

Hierzu 8 Seite(n) Zeichnungen

- Leerseite -

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

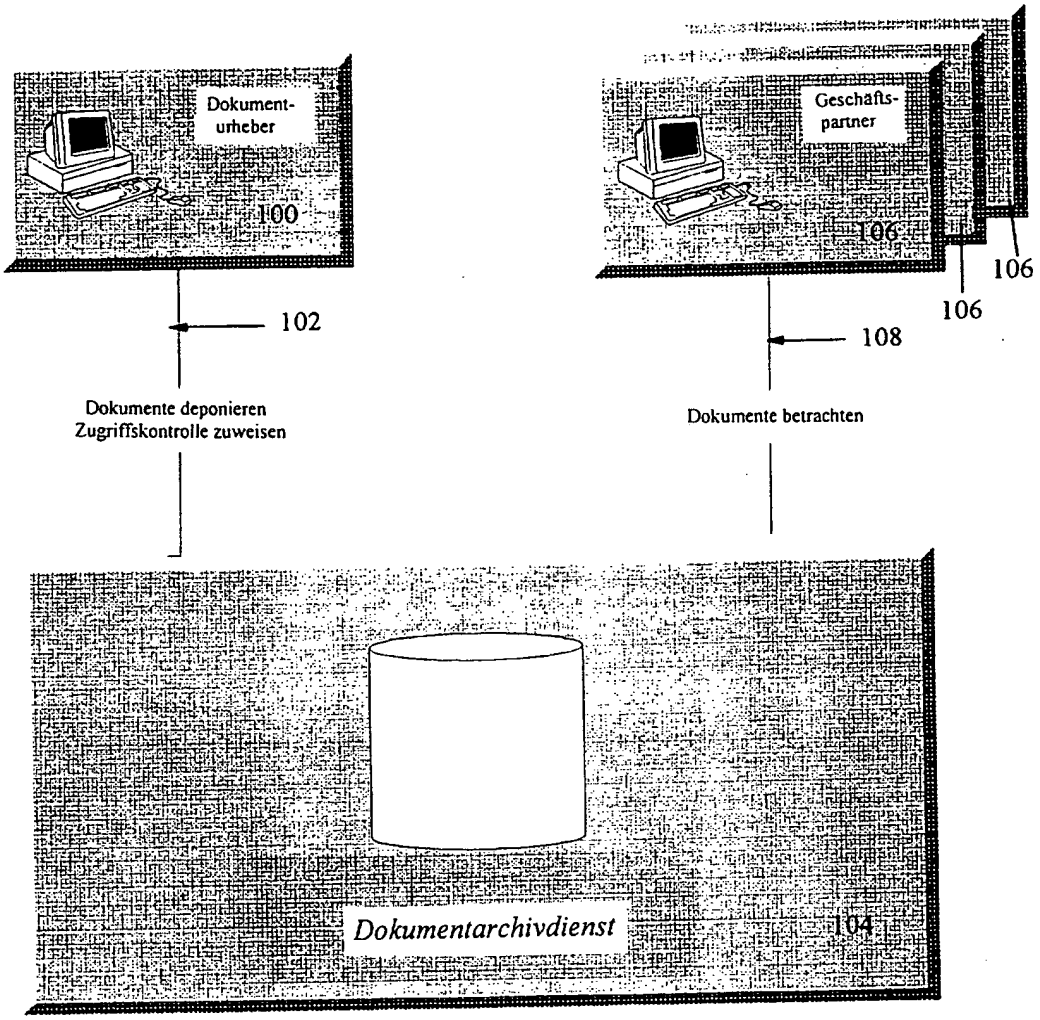


FIG. 1

STAND DER TECHNIK

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

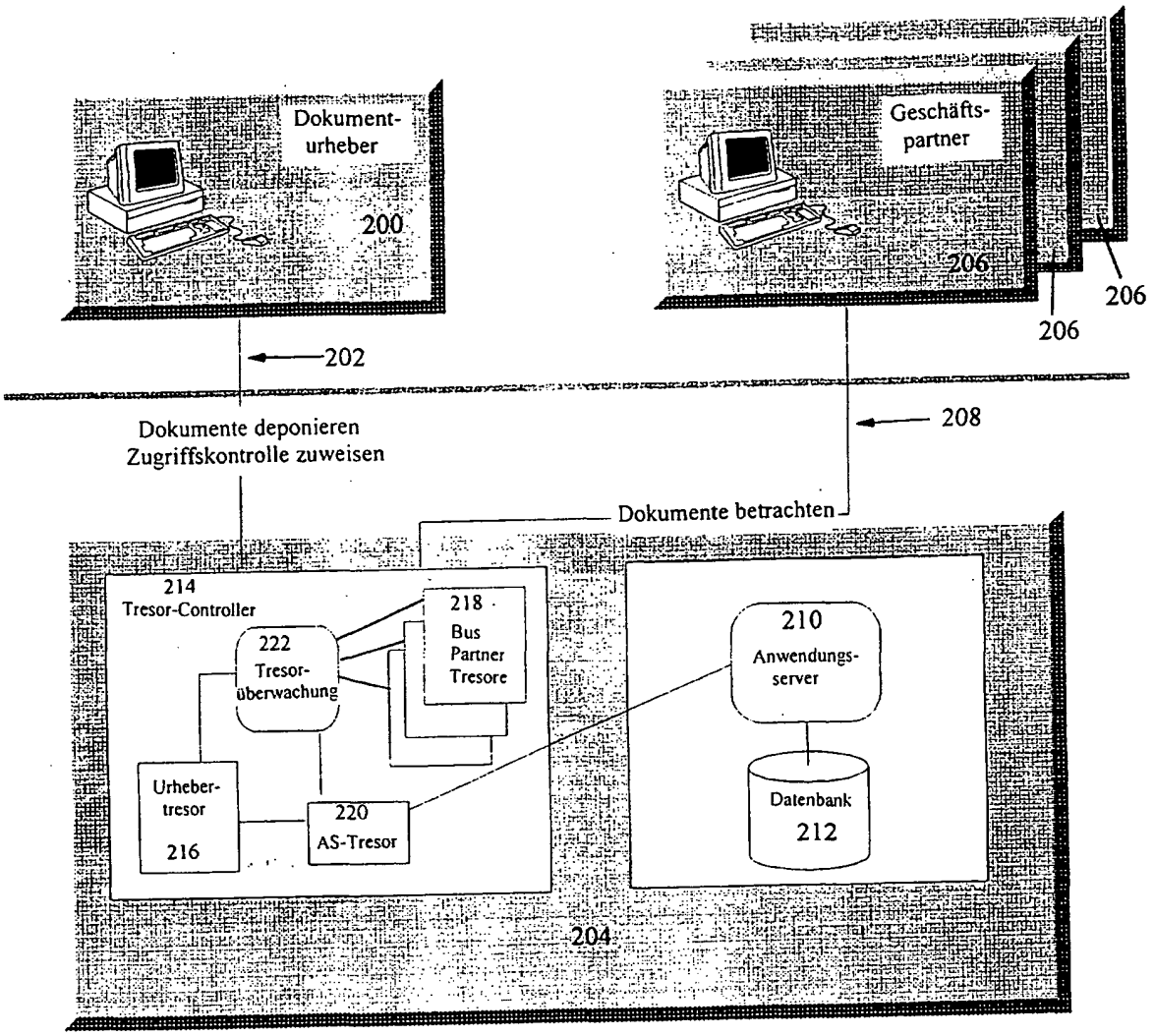


FIG. 2

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

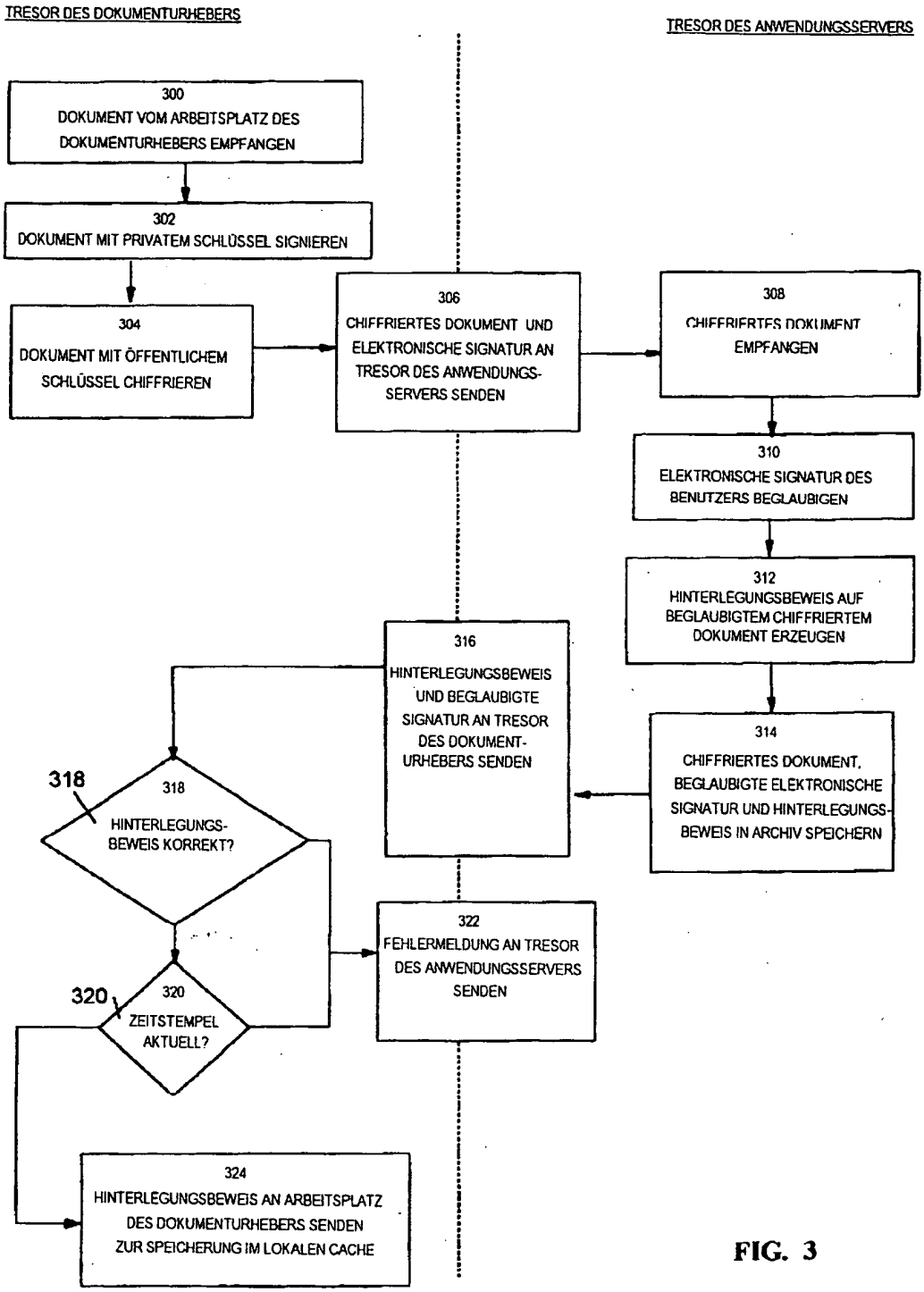


FIG. 3

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

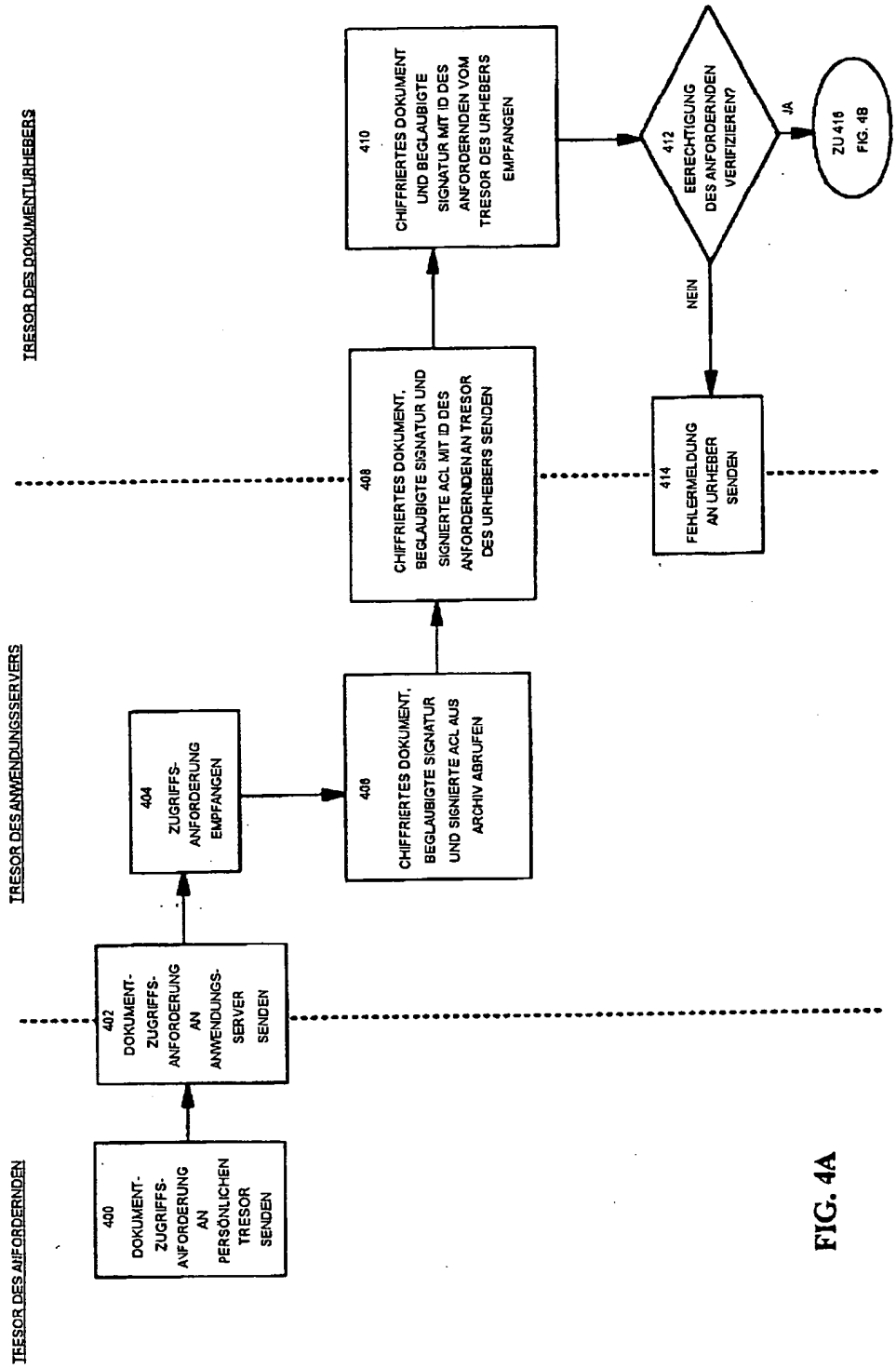


FIG. 4A

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

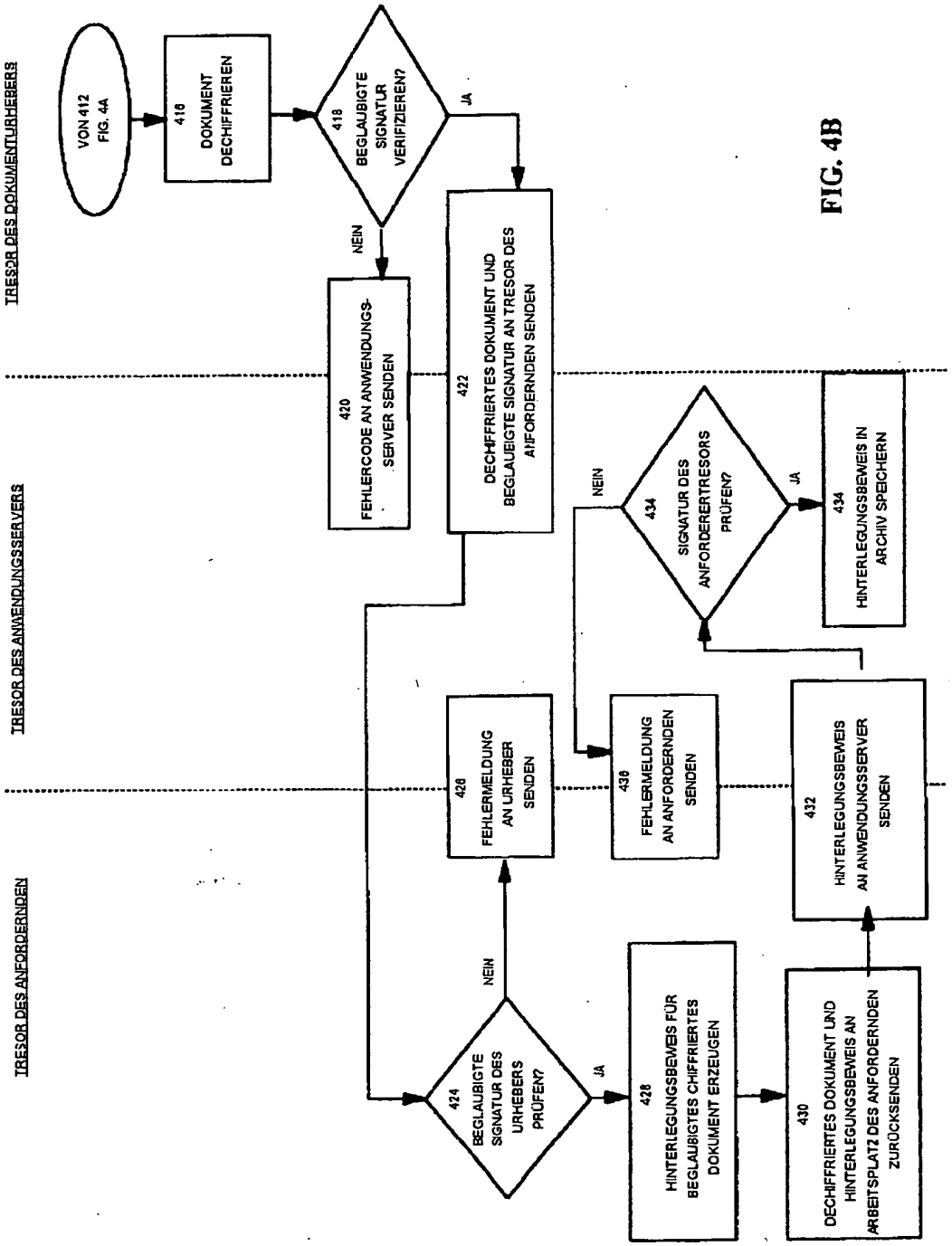


FIG. 4B

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

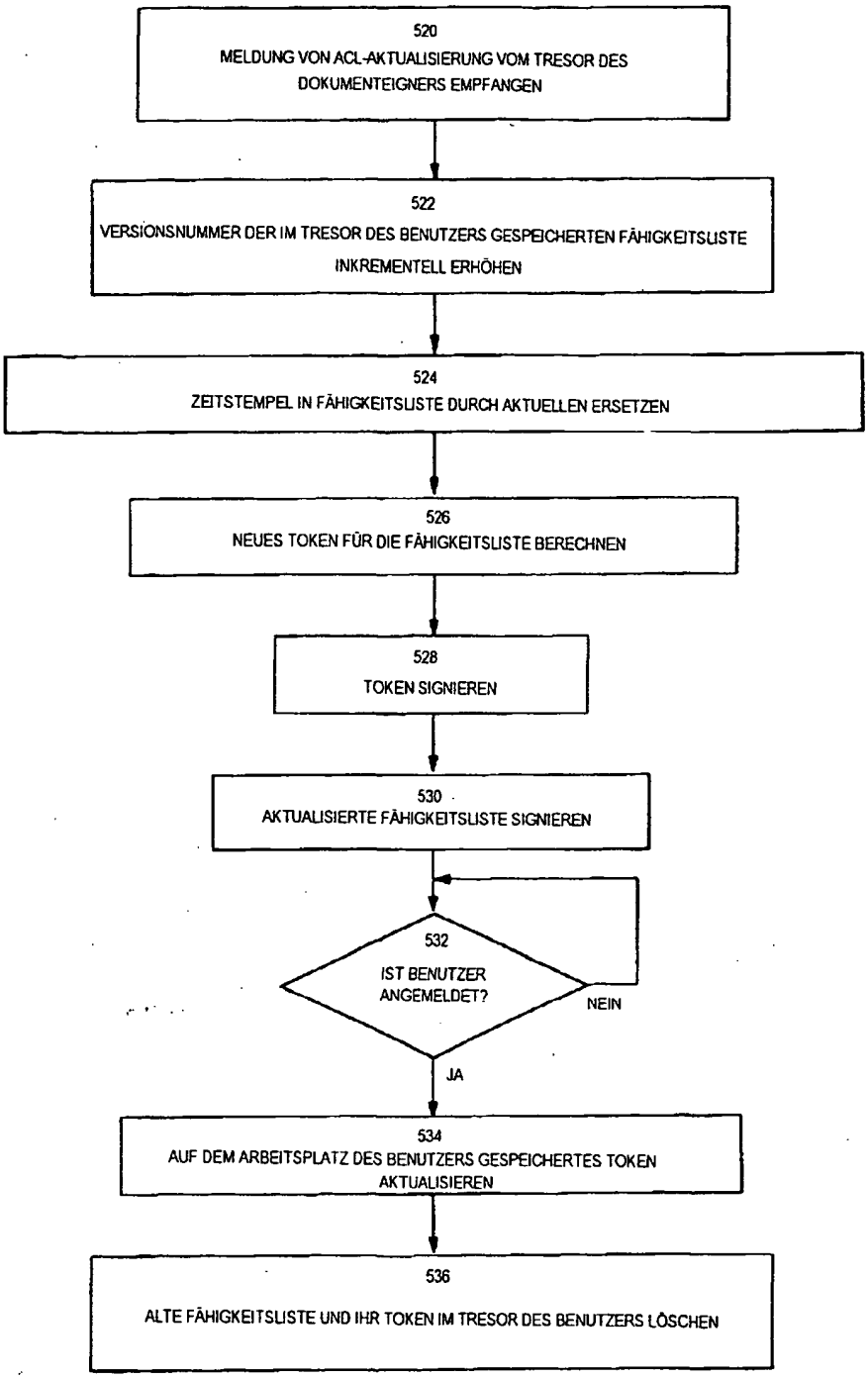
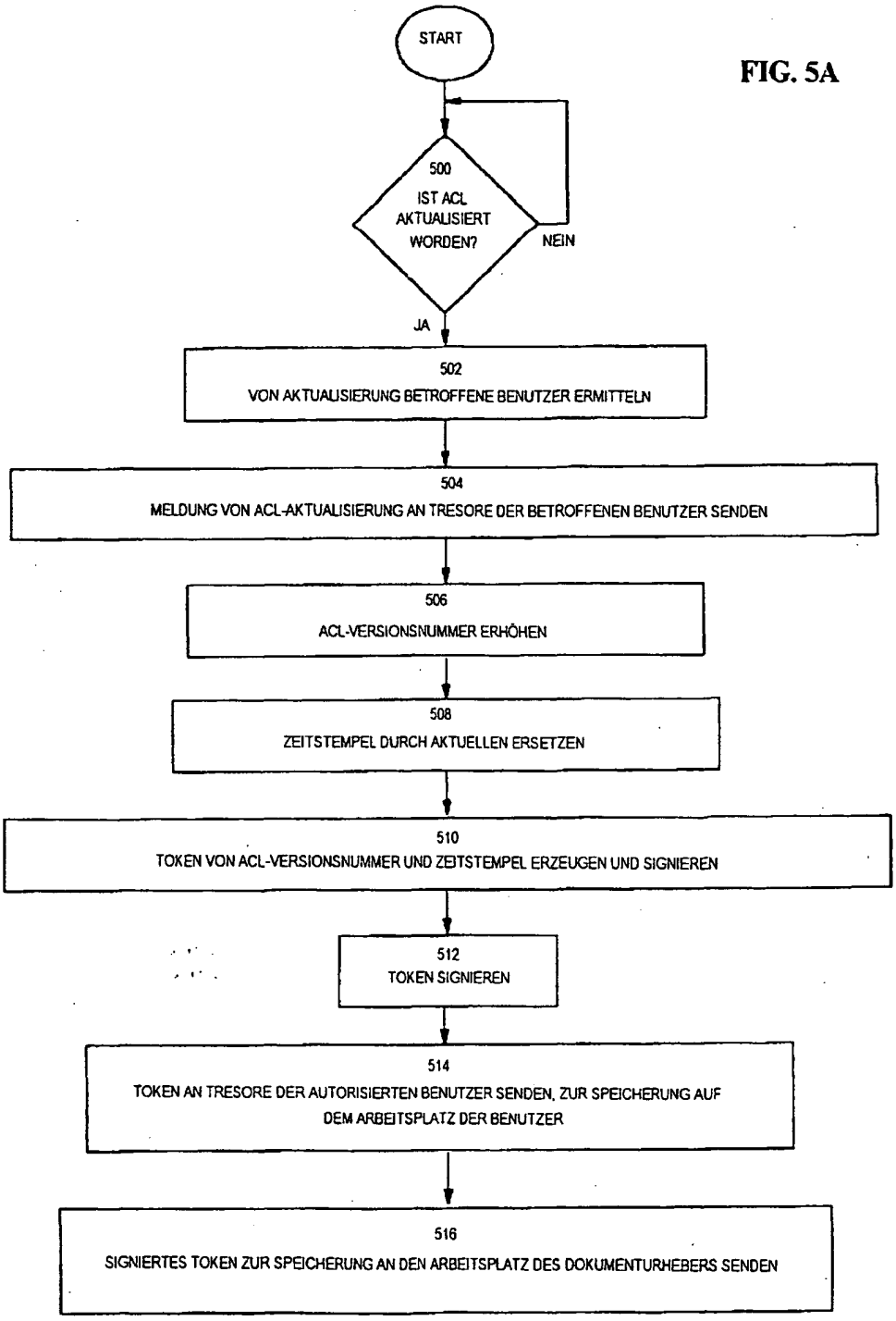


FIG. 5B

Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

FIG. 5A



Nummer: DE 199 60 978 A1
Int. Cl.[7]: G 06 F 17/30
Offenlegungstag: 3. August 2000

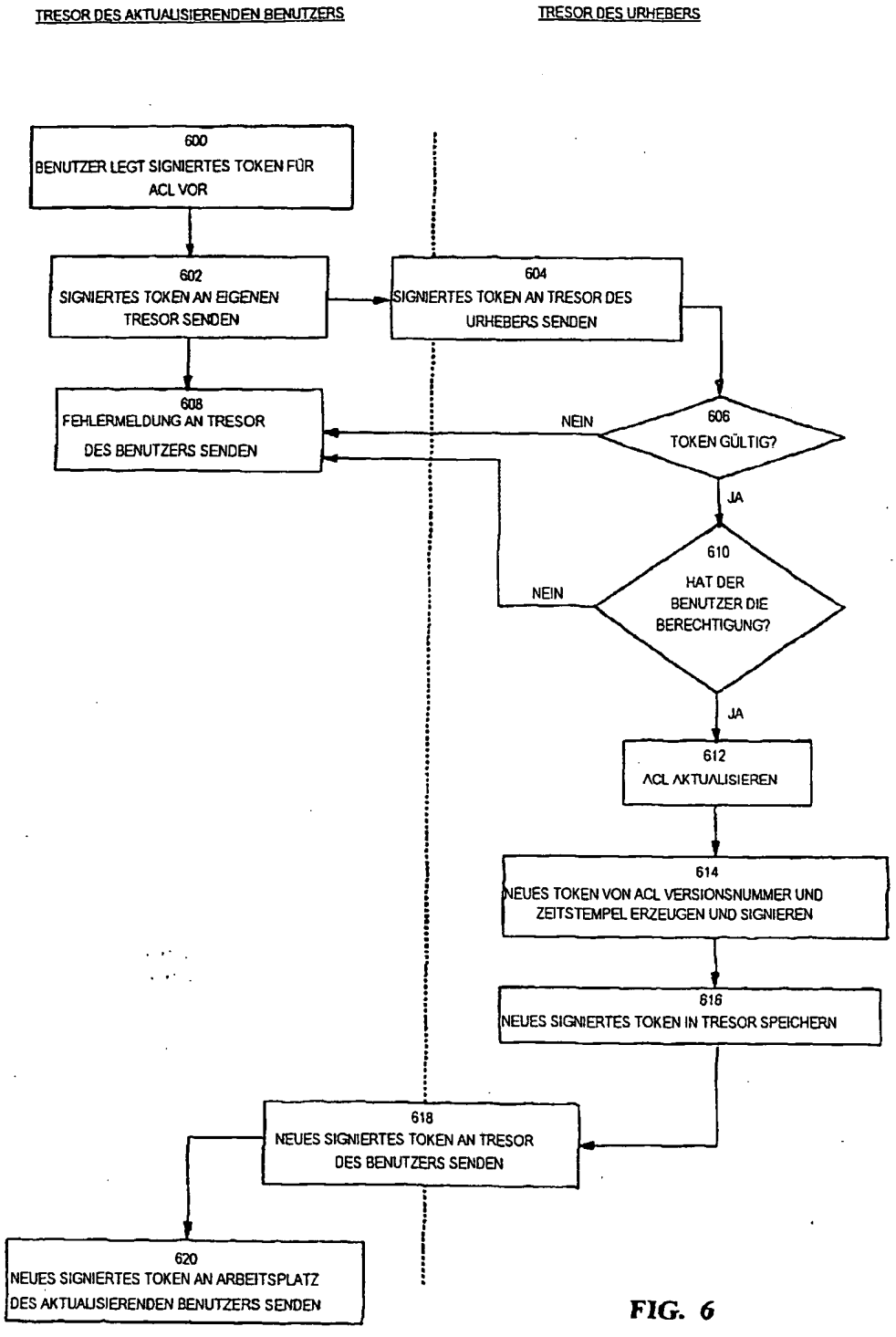


FIG. 6